

## REMARKS

### CLAIM AMENDMENTS

Applicant amends the claims to more particularly point out the subject matter of the invention in view of the Examiner's remarks from the February 23, 2007 office action.

### INTERVIEW SUMMARY

Applicant appreciates the opportunity to discuss the subject matter of the invention in a telephone interview with the Examiner on June 25, 2007. During that interview, both *Kamvysselis* and certain claim amendments were discussed. No agreement was reached.

### SECTION 102 REJECTION OF CLAIMS 2, 7, 8, 9, 15 AND PROGENY

#### Construing "*the data*"

The claimed invention concerns the problem of mirroring certain "selected data" that cannot fit into a slot. In such cases, one must divide the data among several slots, each of which must be mirrored individually.

In many cases, one cannot validate the selected data until all the slots have been copied. In such cases it makes no sense to mirror *any* of the slots until all the slots have been copied, and the data validated.

A conventional mirroring system queues a mirror-request for a slot almost as soon as it receives the slot. Once a valid mirror-request is queued, it is not possible to stop it from being executed. As a result, if the selected data includes three slots of data, and only when the third slot arrives can one discover that the selected data was bad to begin with, the mirror-requests for the first two slots might already be on the mirror-queue. These mirror requests would therefore be executed, whether we like it or not.

The problem addressed by the invention is that of preventing the leading slots from being prematurely mirrored, and doing so without unduly interfering with the normal mirroring process, in which slots are queued for mirroring immediately after they are copied. By holding back the leading slots, one can confirm that the data is valid before queuing mirror requests for any of the slots.

Applicant amends the claims to make clear that the data in the preamble is the union of the constituent portions of the data written to the slots.

It should now be absolutely clear from the amendments to paragraph 1 of claim 2:

1. that "the selected data" from the preamble is too large to fit into just a single slot;
2. that "the selected data" from the preamble must therefore be divided into "portions" that are each small enough to fit into a slot.
3. that each slot ends up storing only a slot-sized portion of the selected data.

#### **Construing "slot"**

The Office suggests that the broadest reasonable interpretation of "slot" is "distinct memory area" and that *Kamysselis* teaches writing portions of data to distinct memory areas. The Office states that in *Kamysselis*, "first and second portions of data may be written to" for example two lines or two blocks of memory.

The Office has made the common mistake of applying the "broadest reasonable interpretation" to the individual words used in a claim. However, Court's oft-quoted phrase "broadest reasonable interpretation" is not a license to deconstruct the claim into its constituent words and to then match each word with a dictionary definition. In fact, the "broadest reasonable interpretation" is to be applied to the claim as a whole. Moreover, the "broadest reasonable interpretation" must be consistent with the specification.

The Office's interpretation of the claim as including the writing of data portions to any "distinct memory area" whatsoever is manifestly inconsistent with the specification.

Applicant has gone through the trouble of carefully describing slots in the specification and describing attributes of slots. After having carefully described slots in detail, Applicant then deliberately chose the word "slot" for the claim. It is improper for the Office to disregard the

specification and to instead re-define slots as simply being any distinct memory areas whatsoever.

Applicant recognizes that one may not import limitations into the claims from the specification. But this is not the case here. Applicant chose "slot" for the claim because the attributes of "slot" were given in the specification.

Moreover, the Office's definition of "slot" as being any "distinct memory area" is so broad as to be absurd. According to the Office's definition, every single one of the bits in a memory must be a "slot" because every single bit is "a distinct memory area."

The Office's definition would also mean that any an eight-bit byte must also be a slot. This leads to the bizarre conclusion that there exist slots that are somehow made up of eight other slots.

The Office's definition would also mean that an entire hard drive is a "slot". After all, a hard drive is a "distinct memory area." Of course, the tracks, cylinders, and sectors of the drive would also become "slots" since they too are "distinct memory areas." The Office's definition of "slot" as being "a distinct memory area" would thus mean that a hard drive is really just a slot that is made of other slots, each of which is itself made of slots.

Nothing in the specification would suggest to one of ordinary skill in the art that "slot," as that term is used in the specification, encompasses bit, byte, word, long word, data block, track, cylinder, sector, hard drive, or magnetic tape.

Indeed, the Office's definition of "slot" as "distinct memory area" is so boundless that even a pair of slots would have to regarded as a slot. After all, a pair of slots forms a memory area that is distinct from all other memory areas, and is therefore well within the Office's proposed definition of "slot."

Under the Office's definition of "slot" as being any "distinct memory area," it is impossible to identify any data storage element whatsoever that *is not* a slot. A definition of a "slot" that encompasses each and every data storage element in existence cannot possibly be a meaningful construction of "slot" as one of ordinary skill in the art would understand that term.

***Kamysselis* fails to teach deleting plural mirror requests**

Claim 2 is amended to recite

"if the selected data....is invalid, deleting the plurality of buffered mirror requests."

Prior to amendment of claim 2, the Office had already conceded that *Kamysselis* failed to teach this limitation, but that it would have been obvious to do delete a plurality of mirror requests because the algorithm of FIG. 6 would be executed more than once.

As amended, it is clear that the mirror requests to be deleted are those for mirroring the constituent portions of the selected data. To the extent *Kamysselis* deletes plural mirror requests through multiple executions of the FIG. 6 algorithm, each such mirror request is associated with a different "selected data." *Kamysselis* does not describe any "selected data" that requires two or more mirror requests to mirror.

Applicant further amends claim 2 to draw attention to the distinction between two states in which a mirror request can exist:

1. a mirror request can be "buffered", or
2. a mirror request can be "sent for execution."

These two states must be construed as distinct, since otherwise, claim 2's recitation of "sending the plurality of buffered mirror requests for execution" would be meaningless.

The above limitation of claim 2 refers only to deleting *buffered* mirror requests, as distinct from mirror requests that have been sent for execution.

In *Kamysselis*, there is no notion of a mirror request being “buffered” as distinct from being “sent for execution.” In *Kamysselis*, there is only one place to store a mirror request: the mirror queue. There is no special location designated for temporarily storing mirror requests so that one can recall them if necessary.

Although one might be tempted to regard the mirror requests waiting in *Kamysselis*'s mirror queues as “buffered” mirror requests, doing so would mean that there would be no mirror requests that have been “sent for execution.”

***Kamysselis* fails to teach buffering mirror requests**

In the previous response Applicant offered an analogy to explain the distinction between buffering and queuing as those terms are used in the specification. Applicant likened the queued mirror requests to a stream of bullets in-flight and the buffered mirror requests to a cartridge full of bullets that have yet to be fired.

The essential distinction Applicant sought to make is that that bullets in-flight cannot be recalled, whereas bullets can easily be removed from the cartridge. Thus, there is obviously a fundamental difference between bullets that are already in-flight and waiting for their turn to strike the target, and bullets that are still in the cartridge waiting to be fired. The only issue, therefore, is what words would make this distinction apparent to one of ordinary skill in the art who has read the specification.

Applicant happened to choose “queued” for the mirror requests that could not be recalled because that word had already been used to in the art for that purpose. Therefore Applicant needed a new word to describe those mirror requests that could still be recalled. Applicant happened to choose “buffered” since that word is commonly used in computer science to indicate temporary storage of data. The usage is consistent throughout the specification. Therefore, one of ordinary skill in the art who has read the specification would have no difficulty understanding the distinction.

The idea of *both* buffering *and* queuing mirror requests is completely absent from *Kanvysseis*. *Kanvysseis* only teaches queuing mirror requests. In particular, *Kanvysseis* fails to teach *buffering* a mirror request as distinct from queuing a mirror request.

Claims 4-6 all depend on claim 2 and are allowable for at least the same reasons set forth above.

Independent claims 7, 8, and 9 recite limitations similar to claim 2 and are allowable for at least the same reasons.

Claims 15 and its dependent claims 17-19 include limitations similar to claims 1 and 4-6 and are therefore patentable for at least the same reasons.

#### **SECTION 101 REJECTION OF CLAIMS 15-19**

The Office rejects claims 15-19 as non-statutory.

Claim 15 recites "[a] computer-readable medium having encoded thereon software for causing storage of data." This is an article of manufacture and is therefore statutory.

Applicant draws attention to *In re Lowry*, 32 F.3d 1579 (CAFC 1994) in which the Federal Circuit reported that the Board reversed the Examiner's rejection of a similar claim under section 101. In that case, the Board said the claim was an article of manufacture.

Applicant also draws attention to *In re Beauregard*, 53 F.3d 1583, 1583-84 (Fed. Cir. 1995), in which the Court approved of the PTO finding that "computer programs embodied in a tangible medium, such as floppy diskettes, are patentable subject matter under 35 U.S.C. § 101".

Applicant also draws attention to the USPTO's *Examination Guidelines for Computer-Related Inventions*, 61 Fed. Reg. 7478, 7482 (1996), which state that "[a] claimed computer-readable medium encoded with a computer program defines structural and functional interrelationships between the computer program and the medium which permit the computer program's functionality to be realized, and is thus statutory."

# SUMMARY

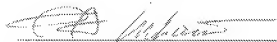
Now pending in this application are claims 2-9 and 15, and 17-19, with claims 2, 7-9, and 15 being independent. Claims 1, 3, 10-14, and 16 are cancelled.

Applicant has advanced only selected arguments, and has done so only for selected claims. This is not, however, to be construed as an admission that no other grounds for patentability of the claims exist.

No additional fees are believed to be due in connection with the filing of this response. Please apply any other charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: June 25, 2007

  
\_\_\_\_\_  
Faustino A. Lichauco  
Reg. No. 41,942

Fish & Richardson P.C.  
225 Franklin Street  
Boston, MA 02110  
Telephone: (617) 542-5070  
Facsimile: (617) 542-8906

21670932.doc